```
---
title: "Assignment6"
author: "Ram"
date: "11/22/2021"
---
```

Question1

Formulate and solve the binary integer programming (BIP) model for this problem using library lpsolve or equivalent in R.
The longest path is the critical path and objective function is given by
Zmax= 3X13 + 5X12 + 3X35 + 2X25 + 2X58 + 4X24 + 6X57 + 4X47 + 1X46 + 7X89 + 4X79 + 5X69
where Xij(i=starting node, j= ending node)

Starting node:
3X13 + 5X12 = 1

Intermediate nodes:
5X12 - 2X25 - 4X24 = 0
3X13 - 3X35 = 0
4X24 - 1X46 - 4X47 = 0
3X35 + 2X25 - 2X58 - 6X57 = 0
1X46 - 5X69 = 0
6X57 + 4X47 - 4X79 = 0
2X58 - 7X89 = 0

Ending node:
7X89 + 4X79 + 5X69 = 1

Where Xij are binary

The longest path is the critical path which is between the nodes (1-2-5-7-9)

````
```{r}
library(lpSolveAPI)
lprec <- make.lp(nrow = 9, ncol = 12)
# nrow is the number of nodes, ncol is the number of arcs
lp.control(lprec, sense="max")

# creating names for arcs and nodes
arc <- c("x12", "x13", "x24", "x25", "x35", "x46", "x47", "x57", "x58",
"x69", "x79", "x89")
node <- c("node1", "node2", "node3", "node4", "node5", "node6", "node7",
"node8", "node9")
# rename the IP object
rownames(lprec) <- node
colnames(lprec) <- arc
# objective function
time <- c(5, 3, 4, 2, 3, 1, 4, 6, 2, 5, 4, 7)
set.objfn(lprec, 1*time)
# node 1 "starting node"
set.row(lprec, 1, c(1,1), indices = c(1,2))
````
````

```
# node 2:8 "intermediate node"
set.row(lprec, 2, c(1,-1,-1), indices = c(1,3,4))
set.row(lprec, 3, c(1,-1), indices = c(2,5))
set.row(lprec, 4, c(1,-1,-1), indices = c(3,6,7))
set.row(lprec, 5, c(1,1,-1,-1), indices = c(4,5,8,9))
set.row(lprec, 6, c(1, -1),indices = c(6,10))
set.row(lprec, 7, c(1,1,-1), indices = c(7,8,11))
set.row(lprec, 8, c(1,-1), indices = c(9,12))
# node 9 "finish node"
set.row(lprec, 9, c(1,1,1), indices = c(10,11,12))
# set constraints type
set.constr.type(lprec, rep("="), 9)
# set constraint RHS
rhs <- c(1, rep(0, 7), 1)
set.rhs(lprec, rhs)
# set all variables type to be binary
set.type(lprec, 1:12, "binary")
write.lp(lprec, "netlp.lp", "lp")
# solve
solve(lprec)

get.objective(lprec)

get.variables(lprec)

get.constraints(lprec)

cbind(arc, get.variables(lprec))
```

So, the critical path which is the maximum objective function we got using lpsolve is 17.

Questin2

1)Determine the maximum return on the portfolio. What is the optimal number of shares to buy for each of the stocks? What is the corresponding dollar amount invested in each stock?
According to the problem returns can be given by
Returns = (Price per share) * (Growth rate of share) + (Dividend per share)

Hence the objective function is
Zmax = 4XS1 + 6.5XS2 + 5.9XS3 + 5.4XH1 + 5.15XH2 + 10XH3 + 8.4XC1 + 6.25XC2

Constraints:

Investment constraint:
40XS1 + 50XS2 + 80XS3 + 60XH1 + 45XH2 + 60XH3 + 30XC1 + 25XC2 <= 2500000

The number of shares invested in any stock must be a multiple of 1000

1000XS1 >= 0; 1000XS2 >= 0; 1000XS3 >= 0
1000XH1 >= 0; 1000XH2 >= 0; 1000XH3 >= 0
1000XC1 >= 0; 1000XC2 >= 0

At least $100,000 must be invested in each of the eight stocks
40XS1 >= 100000; 50XS2 >= 100000; 80XS3 >= 100000;
60XH1 >= 100000; 45XH2 >= 100000; 60XH3 >= 100000;
30XC1 >= 100000; 25XC2 >= 100000

No more than 40 percent of the investment constraints
40XS1 + 50XS2 + 80XS3 <= 1000000
60XH1 + 45XH2 + 60XH3 <= 1000000
30XC1 + 25XC2 <= 1000000

Where XSJ, XHJ, XCJ >= 0 are integers.

```{r}
library(lpSolveAPI)
lprec<-make.lp(0,8)
lp.control(lprec,sense="max")

set.objfn(lprec,c(4,6.5,5.9,5.4,5.15,10,8.4,6.25))

set.type(lprec,c(1:8),type = "integer")

add.constraint(lprec,c(40,50,80,60,45,60,30,25),"<=",2500000,indices =
c(1:8))
add.constraint(lprec,1000,">=",0,indices = 1)
add.constraint(lprec,1000,">=",0,indices = 2)
add.constraint(lprec,1000,">=",0,indices = 3)
add.constraint(lprec,1000,">=",0,indices = 4)
add.constraint(lprec,1000,">=",0,indices = 5)
add.constraint(lprec,1000,">=",0,indices = 6)
add.constraint(lprec,1000,">=",0,indices = 7)
add.constraint(lprec,1000,">=",0,indices = 8)
add.constraint(lprec,40,">=",100000,indices = 1)
add.constraint(lprec,50,">=",100000,indices = 2)
add.constraint(lprec,80,">=",100000,indices = 3)
add.constraint(lprec,60,">=",100000,indices = 4)
add.constraint(lprec,45,">=",100000,indices = 5)
add.constraint(lprec,60,">=",100000,indices = 6)
add.constraint(lprec,30,">=",100000,indices = 7)
add.constraint(lprec,25,">=",100000,indices = 8)
add.constraint(lprec,c(40,50,80),"<=",500000,indices = c(1,2,3))
add.constraint(lprec,c(60,45,60),"<=",1000000,indices = c(4,5,6))
add.constraint(lprec,c(30,25),"<=",1000000,indices = c(7,8))

solve(lprec)

get.objective(lprec)

get.variables(lprec)

get.constraints(lprec)

```
Using lpsolve with integer restriction we get the objective function, maximum returns as 487145.2

number of stocks are:

S1= 2500, S2= 6000, S3= 1250,
H1= 1667, H2= 2223, H3= 3332,
C1= 30000, C2= 4000.

Amount mount invested in each stock:

S1= 100000, S2= 300000, S3= 100000,
H1= 100020, H2= 100035, H3= 799920,
C1= 900000, C2= 100000.

2)Compare the solution in which there is no integer restriction on the number of shares invested. By how much (in percentage terms) do the integer restrictions alter the value of the optimal objective function? By how much (in percentage terms) do they alter the optimal investment quantities?

Using lpsolve without integer restriction we get the objective function,
maximum returns = 487152.8
number of stocks:

S1= 2500.0, S2= 6000.0, S3= 1250.0,
H1= 1667.667, H2= 2222.222, H3= 13333.333,
C1= 30000.0, C2= 4000.0.

The amount invested in each stock

S1= 100000, S2= 300000, S3= 100000,
H1= 100000, H2= 100000, H3= 800000,
C1= 900000, C2= 100000.

The integer restrictions alter the value of the optimal objective function by the percentage 0.00156
Using lpsolve without integer restriction.

```{r}
library(lpSolveAPI)
lprec<-make.lp(0,8)
lp.control(lprec,sense="max")
set.objfn(lprec,c(4,6.5,5.9,5.4,5.15,10,8.4,6.25))
add.constraint(lprec,c(40,50,80,60,45,60,30,25),"<=",2500000,indices =
c(1:8))
add.constraint(lprec,1000,">=",0,indices = 1)
add.constraint(lprec,1000,">=",0,indices = 2)
add.constraint(lprec,1000,">=",0,indices = 3)
add.constraint(lprec,1000,">=",0,indices = 4)
add.constraint(lprec,1000,">=",0,indices = 5)
add.constraint(lprec,1000,">=",0,indices = 6)
add.constraint(lprec,1000,">=",0,indices = 7)
add.constraint(lprec,1000,">=",0,indices = 8)
add.constraint(lprec,40,">=",100000,indices = 1)
add.constraint(lprec,50,">=",100000,indices = 2)
```

```
add.constraint(lprec,80,">=",100000,indices = 3)
add.constraint(lprec,60,">=",100000,indices = 4)
add.constraint(lprec,45,">=",100000,indices = 5)
add.constraint(lprec,60,">=",100000,indices = 6)
add.constraint(lprec,30,">=",100000,indices = 7)
add.constraint(lprec,25,">=",100000,indices = 8)
add.constraint(lprec,c(40,50,80),"<=",1000000,indices = c(1,2,3))
add.constraint(lprec,c(60,45,60),"<=",1000000,indices = c(4,5,6))
add.constraint(lprec,c(30,25),"<=",1000000,indices = c(7,8))


solve(lprec)

get.objective(lprec)

get.variables(lprec)

get.constraints(lprec)
```